File I/O

Working with Excel Files

```
a = round(rand(5,3)*100)
xlswrite('rand.xlsx',a)
b = xlsread('rand.xlsx')
```

Excel files with mixed content

a	123	Cindy
b	333	Suzanne
c	432	David
d	987	Burt

```
>> [nums, txt, raw] = xlsread('texttest.xls')
nums =
 123
 333
 432
 987
txt =
 'a' '' 'Cindy'
 'b' '' 'Suzanne'
 'c' '' 'David'
 'd' '' 'Burt'
```

Save

- save filename
  - Saves all workspace variables
- who -file filename
  - Lists the contents of the file.
- save filename variablename
  - Saves a single variable. You must provide the name of the variable, not its value.
- save -append filename var2
  - Appends another variable to the file.
- save -ascii filename mat
  - Saves in ascii format.

Load

- load filename
  - Loads the contents of the file into current workspace
- load filename var1 var2
  - Loads only var1 and var2 from the file.
- mat = load -ascii filename
  - Loads as an ascii text file.

Higher Level I/O

- load
- save
- type
- input
- disp
- fprintf
- xlsread
- xlswrite

Lower Level I/O

- File identifier
- File offset
- Reading files line by line, or character by character.
  - fread, fgets
  - fgetl
  - fscanf
  - fseek, ftell
  - fwrite

General Steps in File I/O

- open the file
- read from the file, write to the file, or append to the file
- close the file

Opening and closing files

fid = fopen('filename', 'mode');

- If it fails, fid==-1.

- Common modes:
  - r: reading (this is the default)
  - w: writing
  - a: appending
- Closing file:
  - fclose(fid)
  - fclose('all')

File IO

- fid = fopen(filename, mode)

mode	description
'r'	Open file for reading (default).
'w'	Open or create new file for writing. Discard existing contents, if any.
'a'	Open or create new file for writing. Append data to the end of the file.
'r+'	Open file for reading and writing.
'w+'	Open or create new file for reading and writing. Discard existing contents, if any.
'a+'	Open or create new file for reading and writing. Append data to the end of the file.
'A'	Append without automatic flushing. (Used with tape drives.)
'W'	Write without automatic flushing. (Used with tape drives.)

Writing into files

```
>> fid=fopen('temp.txt','a');
>> fprintf(fid, 'this is a string\n');
>> fprintf(fid, '%d %d %d\n', 1,2,3);
>> fclose(fid);
>> edit('temp.txt')
```

Detailed Steps for Reading Files

- Attempt to open the file.
  - Check to ensure that the file open was successful.
- If opened, loop until the end of the file is reached.
  - For each line in the file,
    - read it into a string
    - manipulate the data
- Attempt to close the file.
  - Check to make sure that the file close was successful.

Checking if file was opened

```
fid = fopen('samp.dat');
if fid == -1
        error('File open not successful')
end
% Carry on and use the file!
```

Reading file to the end

```
fid = fopen('filename');
if fid == -1; error('File open not successful'); end

while feof(fid) == 0
        % Read one line into a string variable
        aline = fgetl(fid);
        if ~ischar(aline); break; end

        % Use string functions to extract numbers, strings,
        % etc. from the line
        % Do something with the data!
end
fclose(fid);
```

Exercise

- Write a function that reads in a given file and returns the number of times the lower case letter 'a' appears in the file.

Left here.

Exercise

- Write a function pdb_coords.m that extracts all of the atom coordinates from a given pdb file and returns them as an n-by-3 matrix M. Test the file using:

```
>> M=pdb_coords('1crn.pdb')
```

ATOM entry lines is formatted such that the characters 13-16, **31-38, 39-46, 47-54**, and 61-66 contain the atom type (e.g., "CA", ignoring any spaces), the x,y,z coordinates, and the B-factor of an atom, respectively.

Exercise

- Write a function that reads a text file containing a number followed by a letter on each line, and returns the sum of the numbers. Use strsplit to solve this problem. Then change your code to solve the problem using sscanf.
- Example file contents:

5.3 a

2.25 b

3.3 c

Formatted Reading Into a Matrix

- mat = fscanf(fid, 'format', [dimensions])
  - fscanf reads the file into mat column-wise. Each parsed string is read into a different column.
- mat = fscanf(fid,'%f %c',[2 inf])
- mat = fscanf(fid,'%f%c',[2 inf])

Formatted Reading Into a CellArray

- cellarray = textscan(fid, 'format');
- subjdata = textscan(fid,'%f %c');

Exercise

- Write a function that reads the following file and plot x vs. y.

- Use fgetl, fscanf, and textscan to solve this problem.

Writing to files

- fprintf(fid, 'format', variable(s));

- Exercise: Create a random 10x10 matrix. Write this matrix into a text file, each line containing a row of the matrix, with numbers separated by commas.
- Exercise: Write a function that reads the text file you just wrote.